

Title: IMPLEMENTING SDN FIREWALL USING MININET

Presented By

124157077, YANAMADALA UJJWALA,

Guide Name: Dr. R. Kavitha, CSE(CSBC)/SoC, SASTRA University

AIM & OBJECTIVES

The aim of the project is implementing the firewall, using SDN Network in the layer 2,3 & 4.

Objective 1: understanding SDN networking

Objective 2: Installing firewall rules

Objective 3: Benefits of having programmable controller

Sample problem statements are:

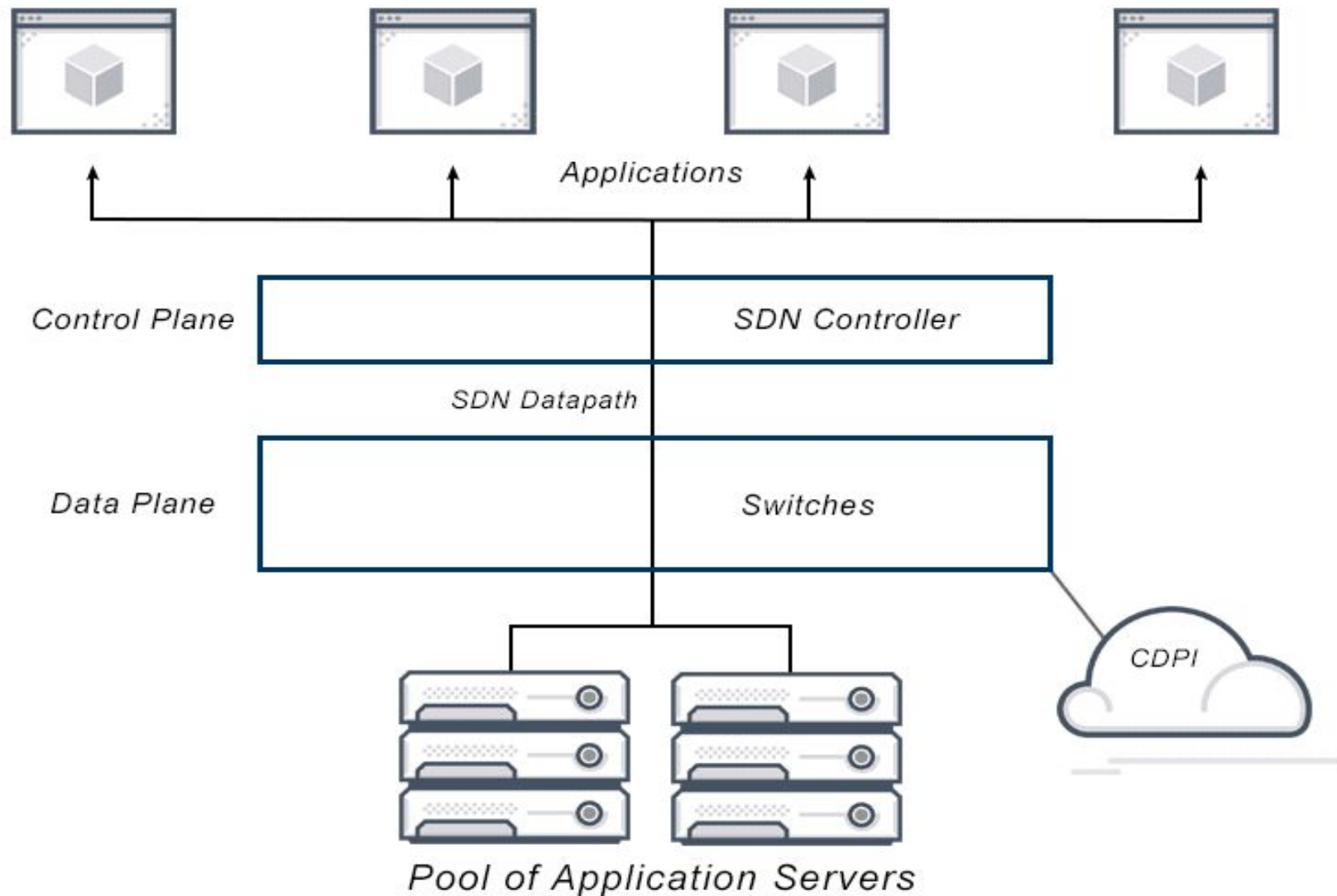
1. Unwanted network traffic due to traditional firewalls and replacing costly layer 4,7 firewalls

The Solutions are:

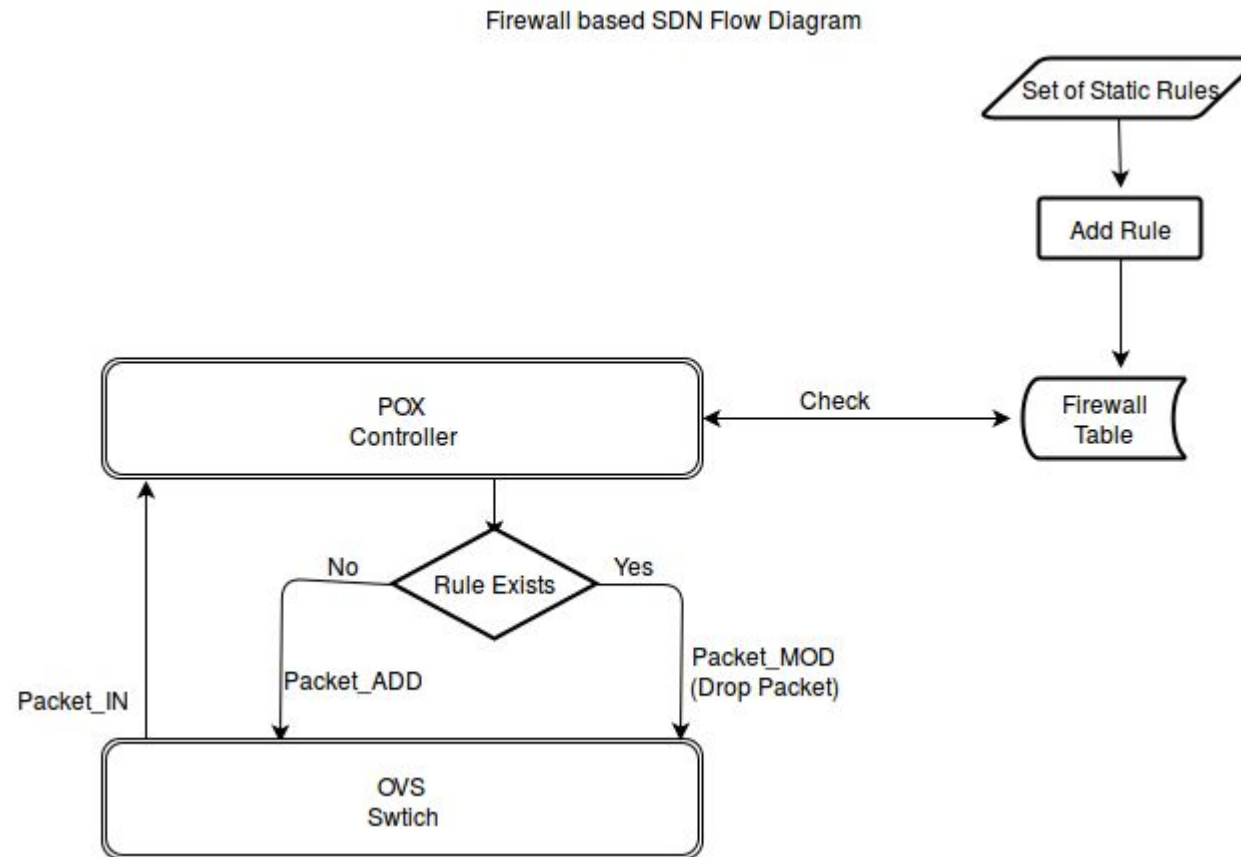
1. Unwanted network traffic - SDN implementation in layer 2,3&4 of OSI

- SDN separates the network control plane from the data plane and gives it to a separate piece of software called controller that runs at the control plane.
- SDN divides the control plane—which chooses how to manage the traffic—from the devices that make up the data plane and send traffic in accordance with those decisions.
- Between the control layer and data layer of an SDN architecture, OpenFlow is the first and current industrially standardized SDN protocol.

Software Defined Networking (SDN)



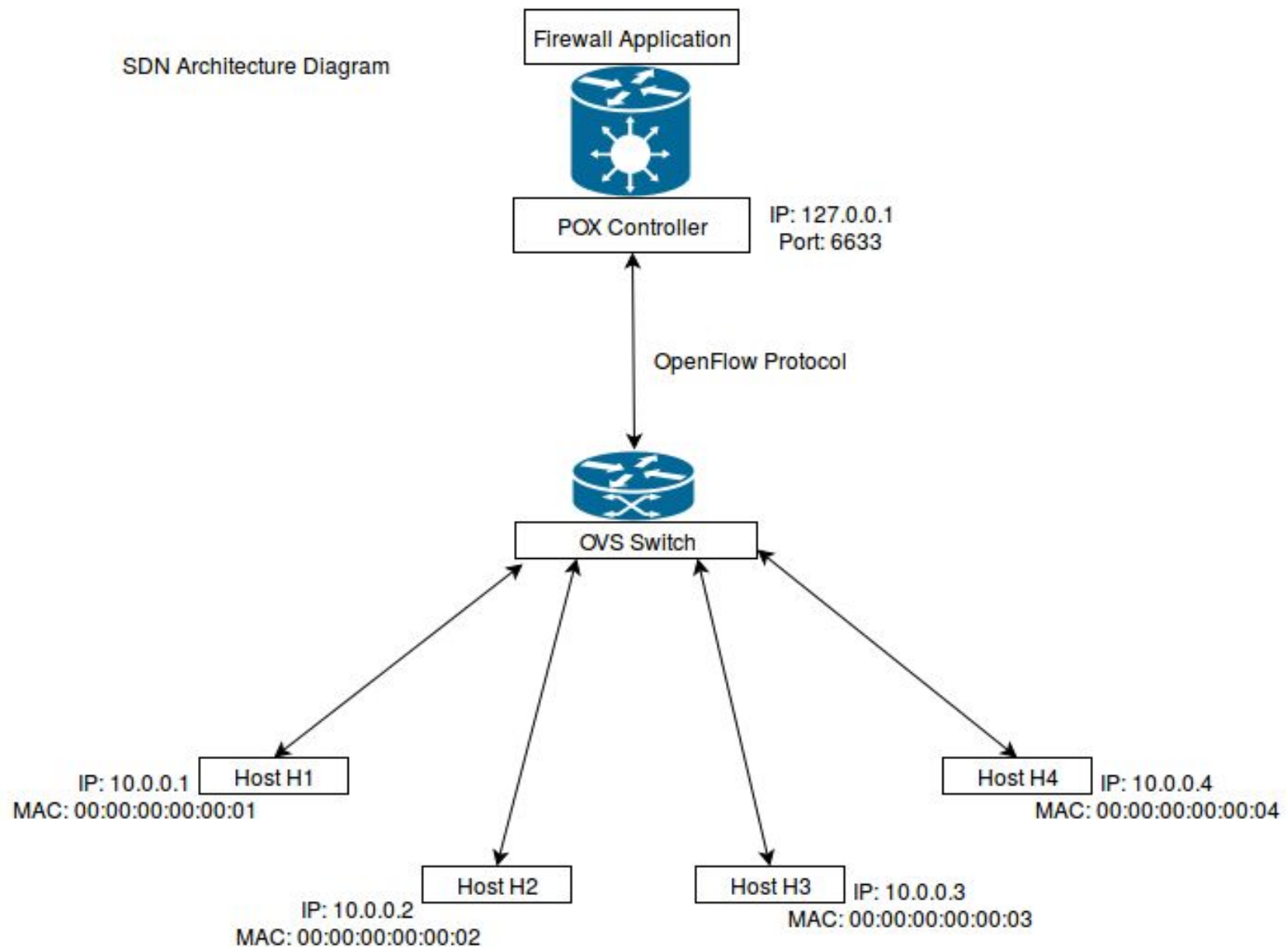
- Firewall's primary duty is to filter packets based on their parsed headers, then permit or refuse them in accordance with predefined rules.



Testing Firewall

- MAC rule: it represents the Layer 2 firewall which catches Ethernet header to check if it belongs to host1 (00:00:00: 00:00:01), it should be blocked in both directions.
- IP rule: it represents the Layer 3 firewall which detects IP packet to check if it matches the rule, then it should not be allowed to forward; and here we specified blocking IP Host-to-Host connectivity.
- TCP and UDP rule: our firewall works here to catch Layer 4 traffic to perform port security. Once it catches TCP segment or UDP datagram, the rules are executed accordingly.

Network Architecture



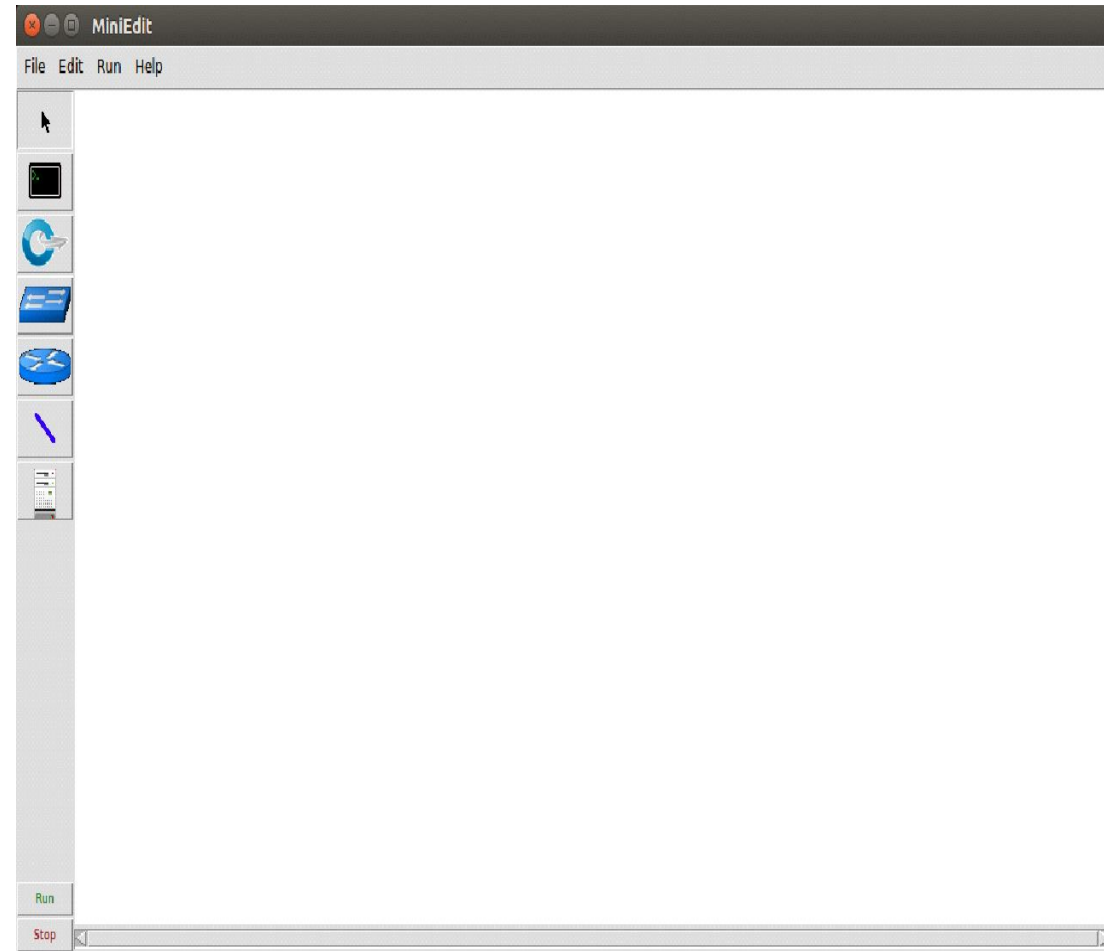
Sample I/O screen shots

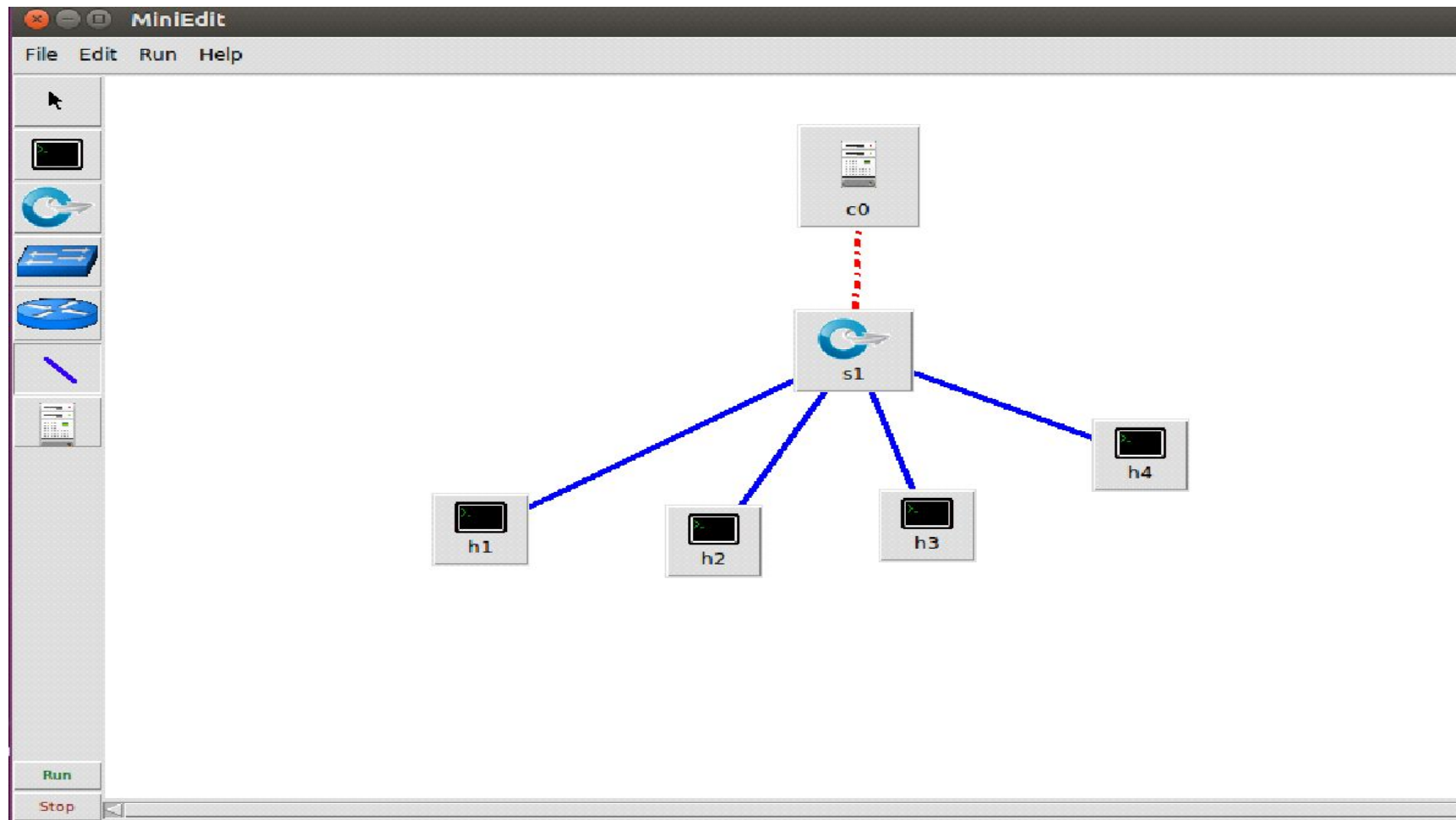
```
admin123@admin123-VirtualBox:~$ cd pox
admin123@admin123-VirtualBox:~/pox$ ./pox.py log.level --INFO poxController_firewall
POX 0.7.0 (gar) / Copyright 2011-2020 James McCauley, et al.
WARNING:version:POX requires one of the following versions of Python: 3.6 3.7 3.8 3.9
WARNING:version:You're running Python 3.5.
WARNING:version:If you run into problems, try using a supported version.
INFO:core:POX 0.7.0 (gar) is up.
INFO:openflow.of_01:[00-00-00-00-00-07 1] connected
```

```
admin123@admin123-VirtualBox: ~
admin123@admin123-VirtualBox:~$ sudo ~/mininet/examples/miniedit.py
[sudo] password for admin123:
```

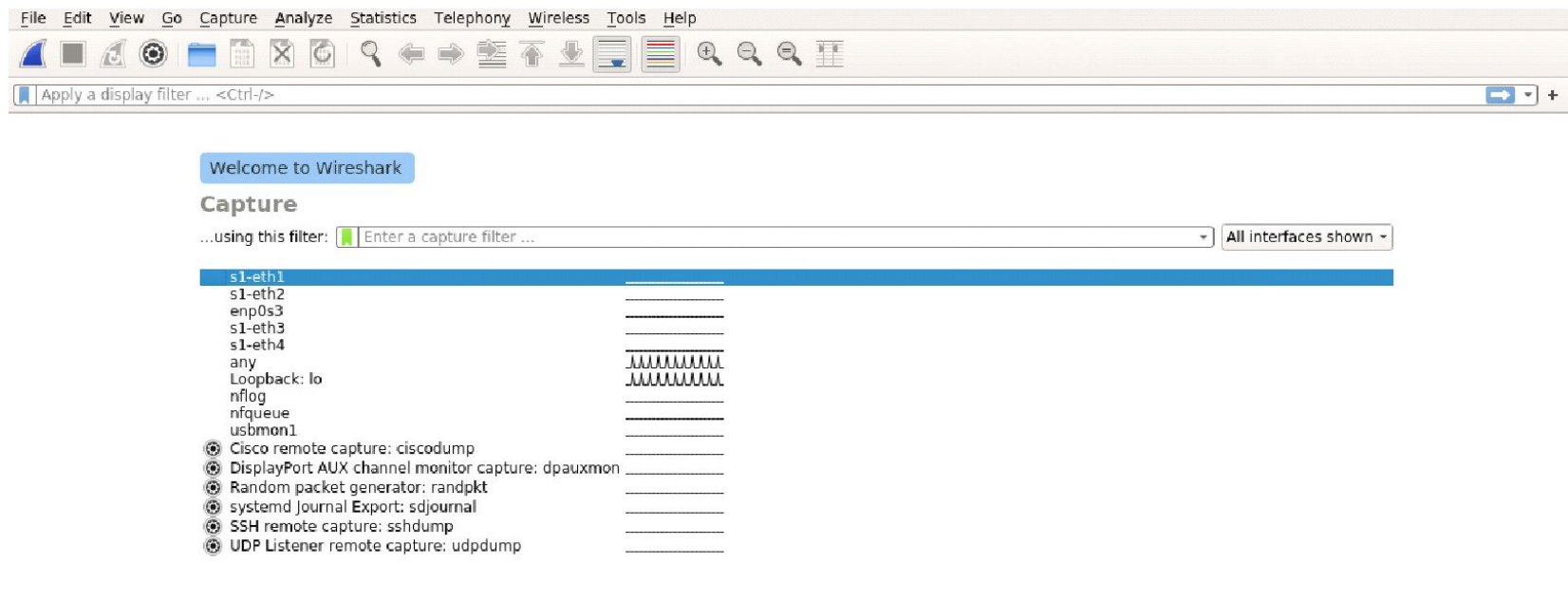


```
root@admin123-VirtualBox: /home/admin123
admin123@admin123-VirtualBox: ~/pox$ cd
admin123@admin123-VirtualBox: ~$ sudo su
[sudo] password for admin123:
root@admin123-VirtualBox: /home/admin123# python mininetScript.py
*** Creating network
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c
*** Starting 1 switches
s1 ...
Dumping host connections
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s1-eth3
h4 h4-eth0:s1-eth4
*** Starting CLI:
mininet> 
```





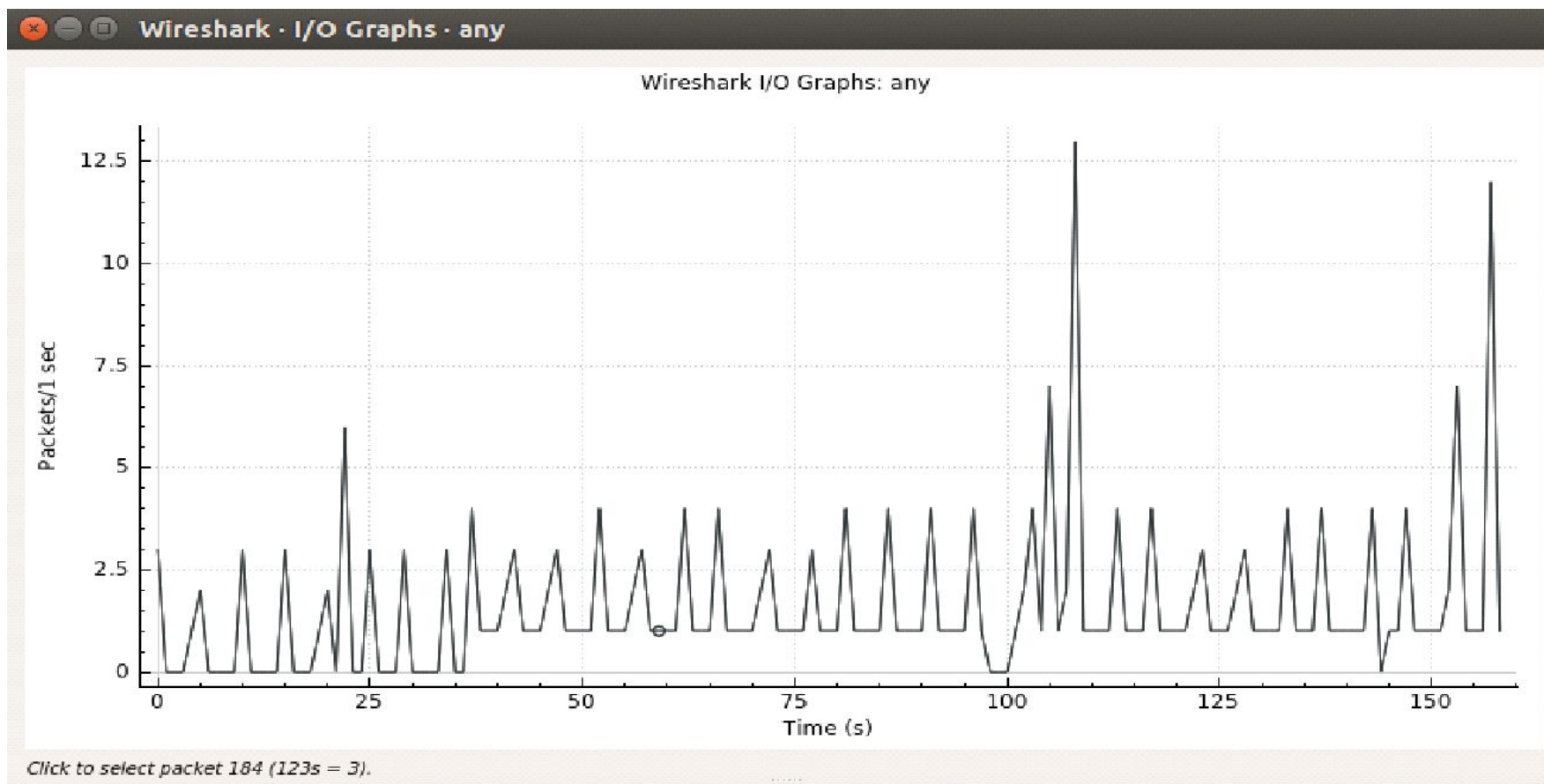
```
root@admin123-VirtualBox:/home/admin123# wireshark
```

```
mininet> h1 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
^C
--- 10.0.0.4 ping statistics ---
92 packets transmitted, 0 received, 100% packet loss, time 93166ms
```



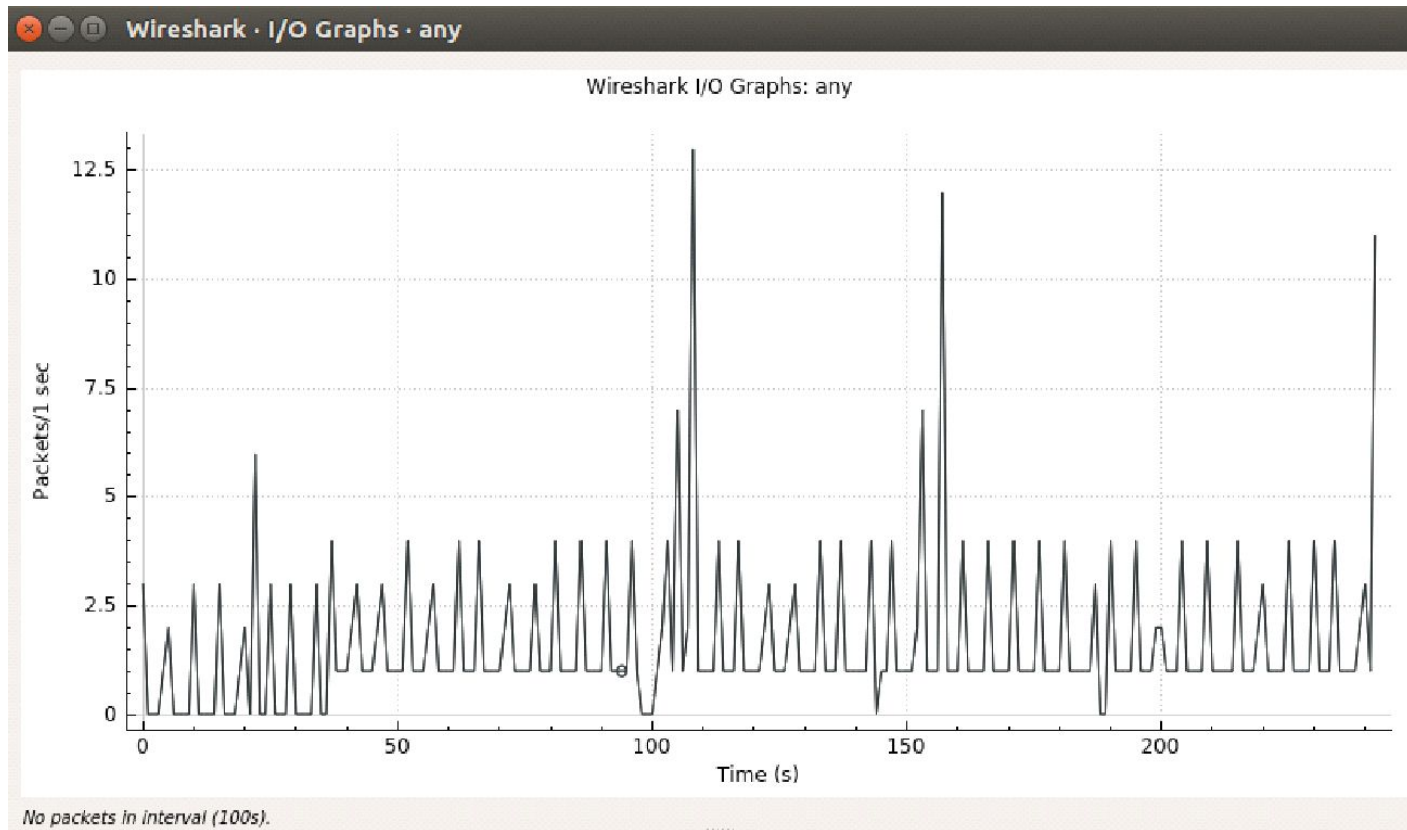
39	41.999775217	127.0.0.1	127.0.0.1	OpenFl...	76 Type: OFPT_ECHO_REQUEST
40	42.000114826	127.0.0.1	127.0.0.1	OpenFl...	76 Type: OFPT_ECHO_REPLY
41	42.000121307	127.0.0.1	127.0.0.1	TCP	68 35716 → 6633 [ACK] Seq=309 Ack=145 Win=342 Len=0 TSval=134540598 TSecr=134540598
42	42.190964115	00:00:00_00:00:02		ARP	44 Who has 10.0.0.4? Tell 10.0.0.2
43	43.215377098	00:00:00_00:00:02		ARP	44 Who has 10.0.0.4? Tell 10.0.0.2
44	44.239432967	00:00:00_00:00:02		ARP	44 Who has 10.0.0.4? Tell 10.0.0.2
45	45.263029512	00:00:00_00:00:02		ARP	44 Who has 10.0.0.4? Tell 10.0.0.2
46	46.287761179	00:00:00_00:00:02		ARP	44 Who has 10.0.0.4? Tell 10.0.0.2
47	46.999735461	127.0.0.1	127.0.0.1	OpenFl...	76 Type: OFPT_ECHO_REQUEST
48	47.000002524	127.0.0.1	127.0.0.1	OpenFl...	76 Type: OFPT_ECHO_REPLY
49	47.000010172	127.0.0.1	127.0.0.1	TCP	68 35716 → 6633 [ACK] Seq=317 Ack=153 Win=342 Len=0 TSval=134545598 TSecr=134545598
50	47.311647919	00:00:00_00:00:02		ARP	44 Who has 10.0.0.4? Tell 10.0.0.2





```
mininet> h2 ping h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
From 10.0.0.2 icmp_seq=1 Destination Host Unreachable
From 10.0.0.2 icmp_seq=2 Destination Host Unreachable
From 10.0.0.2 icmp_seq=3 Destination Host Unreachable
From 10.0.0.2 icmp_seq=4 Destination Host Unreachable
From 10.0.0.2 icmp_seq=5 Destination Host Unreachable
From 10.0.0.2 icmp_seq=6 Destination Host Unreachable
From 10.0.0.2 icmp_seq=7 Destination Host Unreachable
From 10.0.0.2 icmp_seq=8 Destination Host Unreachable
From 10.0.0.2 icmp_seq=9 Destination Host Unreachable
From 10.0.0.2 icmp_seq=10 Destination Host Unreachable
From 10.0.0.2 icmp_seq=11 Destination Host Unreachable
From 10.0.0.2 icmp_seq=12 Destination Host Unreachable
^C
--- 10.0.0.4 ping statistics ---
15 packets transmitted, 0 received, +12 errors, 100% packet loss, time 14332ms
pipe 4
mininet>
```

160	108.2828/1020	127.0.0.1	127.0.0.1	ICMP	68 35/16 → 6633 [ACK] Seq=649 Ack=481 Win=342 Len=0 ISval=134606881 ISecr=134606883
161	109.167644817	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x6d95, seq=7/1792, ttl=64 (no response found!)
162	110.191430026	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x6d95, seq=8/2048, ttl=64 (no response found!)
163	111.214912404	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x6d95, seq=9/2304, ttl=64 (no response found!)
164	112.238941412	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x6d95, seq=10/2560, ttl=64 (no response found!)
165	113.000301850	127.0.0.1	127.0.0.1	OpenFl...	76 Type: OFPT_ECHO_REQUEST
166	113.000532590	127.0.0.1	127.0.0.1	OpenFl...	76 Type: OFPT_ECHO_REPLY
167	113.000538703	127.0.0.1	127.0.0.1	TCP	68 35716 → 6633 [ACK] Seq=657 Ack=489 Win=342 Len=0 TSval=134611598 TSecr=134611599
168	113.262943980	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x6d95, seq=11/2816, ttl=64 (no response found!)
169	114.286915081	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x6d95, seq=12/3072, ttl=64 (no response found!)
170	115.311158986	10.0.0.1	10.0.0.4	ICMP	100 Echo (ping) request id=0x6d95, seq=13/3328, ttl=64 (no response found!)



```
mininet> xterm h1
mininet> xterm h3
mininet> xterm h3
mininet> xterm h1
```



```
"Node: h3"
root@admin123-VirtualBox:/home/admin123# iperf -s -p 22 &
[1] 27792
root@admin123-VirtualBox:/home/admin123# -----
Server listening on TCP port 22
TCP window size: 85,3 KByte (default)
-----
█
```

```
"Node: h3"
root@admin123-VirtualBox:/home/admin123# iperf -s -p 80 &
[1] 27884
root@admin123-VirtualBox:/home/admin123# -----
Server listening on TCP port 80
TCP window size: 85,3 KByte (default)
-----
█
```

```
"Node: h1"
root@admin123-VirtualBox:/home/admin123# iperf -c h3 -p 22 -t 2 -i 1
error: Temporary failure in name resolution
root@admin123-VirtualBox:/home/admin123# █
```

```
"Node: h1"
root@admin123-VirtualBox:/home/admin123# iperf -c h3 -p 80 -t 2 -i 1
error: Temporary failure in name resolution
root@admin123-VirtualBox:/home/admin123# █
```


- This firewall does not maintain track of the connection's state, rendering this work stateless, which is a limitation.
- In order to decrease the burden on the controller in the event of high traffic, we will also improve our firewall further so that it may be spread.
- Due to centralized controller , the cost of damage would me great in SDN.

Thank You